



CHAPTER 9

Configuring Interface Characteristics

This chapter describes the types of interfaces on a Catalyst 3550 switch and how to configure them. The chapter has these sections:

- [Understanding Interface Types, page 9-1](#)
- [Using the Interface Command, page 9-9](#)
- [Configuring Ethernet Interfaces, page 9-13](#)
- [Configuring Layer 3 Interfaces, page 9-19](#)
- [Monitoring and Maintaining the Interfaces, page 9-21](#)



Note

For complete syntax and usage information for the commands used in this chapter, see the switch command reference for this release and the online *Cisco IOS Interface Command Reference, Release 12.2*.

Understanding Interface Types

This section describes the different types of interfaces supported by the switch with references to chapters that contain more detailed information about configuring these interface types. The rest of the chapter describes configuration procedures for physical interface characteristics.

These sections are included:

- [Port-Based VLANs, page 9-2](#)
- [Switch Ports, page 9-2](#)
- [Switch Virtual Interfaces, page 9-4](#)
- [Routed Ports, page 9-4](#)
- [EtherChannel Port Groups, page 9-5](#)
- [Power Over Ethernet Ports, page 9-5](#)
- [Connecting Interfaces, page 9-7](#)

Port-Based VLANs

A VLAN is a switched network that is logically segmented by function, team, or application, without regard to the physical location of the users. For more information about VLANs, see [Chapter 11, “Configuring VLANs.”](#) Packets received on a port are forwarded only to ports that belong to the same VLAN as the receiving port. Network devices in different VLANs cannot communicate with one another without a Layer 3 device to route traffic between the VLANs.

VLAN partitions provide hard firewalls for traffic in the VLAN, and each VLAN has its own MAC address table. A VLAN comes into existence when a local port is configured to be associated with the VLAN, when the VLAN Trunking Protocol (VTP) learns of its existence from a neighbor on a trunk, or when a user creates a VLAN.

To configure normal-range VLANs (VLAN IDs 1 to 1005), use the **vlan *vlan-id*** global configuration command to enter config-vlan mode or the **vlan database** privileged EXEC command to enter VLAN configuration mode. The VLAN configurations for VLAN IDs 1 to 1005 are saved in the VLAN database. To configure extended-range VLANs (VLAN IDs 1006 to 4094), you must use config-vlan mode with VTP mode set to transparent. Extended-range VLANs are not added to the VLAN database. When VTP mode is transparent, the VTP and VLAN configuration is saved in the switch running configuration, and you can save it in the switch startup configuration file by entering the **copy running-config startup-config** privileged EXEC command.

Add ports to a VLAN by using the **switchport** interface configuration commands:

- Identify the interface.
- For a trunk port, set trunk characteristics, and if desired, define the VLANs to which it can belong.
- For an access port, set and define the VLAN to which it belongs.
- For a tunnel port, set and define the VLAN ID for the customer-specific VLAN tag. See [Chapter 14, “Configuring IEEE 802.1Q and Layer 2 Protocol Tunneling.”](#)

Switch Ports

Switch ports are Layer 2-only interfaces associated with a physical port. A switch port can be an access port, a trunk port, or a tunnel port. You can configure a port as an access port or trunk port or let the Dynamic Trunking Protocol (DTP) operate on a per-port basis to determine if a switch port should be an access port or a trunk port by negotiating with the port on the other end of the link. You must manually configure tunnel ports as part of an asymmetric link connected to an IEEE 802.1Q trunk port. Switch ports are used for managing the physical interface and associated Layer 2 protocols and do not handle routing or bridging.

Configure switch ports by using the **switchport** interface configuration commands. For detailed information about configuring access port and trunk port characteristics, see [Chapter 11, “Configuring VLANs.”](#) For more information about tunnel ports, see [Chapter 14, “Configuring IEEE 802.1Q and Layer 2 Protocol Tunneling.”](#)

Access Ports

An access port belongs to and carries the traffic of only one VLAN (unless it is configured as a voice VLAN port). Traffic is received and sent in native formats with no VLAN tagging. Traffic arriving on an access port is assumed to belong to the VLAN assigned to the port. If an access port receives a tagged

packet (Inter-Switch Link [ISL] or IEEE 802.1Q tagged) for the VLAN assigned to the port, the packet is forwarded. If the port receives a tagged packet for another VLAN, the packet is dropped, the source address is not learned, and the frame is counted in the *No destination* statistic.

Two types of access ports are supported:

- Static access ports are manually assigned to a VLAN.
- VLAN membership of dynamic access ports is learned through incoming packets. By default, a dynamic access port is not a member of any VLAN, and forwarding to and from the port is enabled only when the VLAN membership of the port is discovered. Dynamic access ports on the switch are assigned to a VLAN by a VLAN Membership Policy Server (VMPS). The VMPS can be a Catalyst 6000 series switch; the Catalyst 3550 switch does not support the function of a VMPS.

You can also configure an access port with an attached Cisco IP Phone to use one VLAN for voice traffic and another VLAN for data traffic from a device attached to the phone. For more information about voice VLAN ports, see [Chapter 13, “Configuring Voice VLAN.”](#)

Trunk Ports

A trunk port carries the traffic of multiple VLANs and by default is a member of all VLANs in the VLAN database. Two types of trunk ports are supported:

- In an ISL trunk port, all received packets are expected to be encapsulated with an ISL header, and all transmitted packets are sent with an ISL header. Native (non-tagged) frames received from an ISL trunk port are dropped.
- An IEEE 802.1Q trunk port supports simultaneous tagged and untagged traffic. An IEEE 802.1Q trunk port is assigned a default Port VLAN ID (PVID), and all untagged traffic travels on the port default PVID. All untagged traffic and tagged traffic with a NULL VLAN ID are assumed to belong to the port default PVID. A packet with a VLAN ID equal to the outgoing port default PVID is sent untagged. All other traffic is sent with a VLAN tag.

Although by default, a trunk port is a member of every VLAN known to the VTP, you can limit VLAN membership by configuring an allowed list of VLANs for each trunk port. The list of allowed VLANs does not affect any other port but the associated trunk port. By default, all possible VLANs (VLAN ID 1 to 4094) are in the allowed list. A trunk port can only become a member of a VLAN if VTP knows of the VLAN and the VLAN is in the enabled state. If VTP learns of a new, enabled VLAN and the VLAN is in the allowed list for a trunk port, the trunk port automatically becomes a member of that VLAN and traffic is forwarded to and from the trunk port for that VLAN. If VTP learns of a new, enabled VLAN that is not in the allowed list for a trunk port, the port does not become a member of the VLAN, and no traffic for the VLAN is forwarded to or from the port.

For more information about trunk ports, see [Chapter 11, “Configuring VLANs.”](#)

Tunnel Ports

Tunnel ports are used in IEEE 802.1Q tunneling to segregate the traffic of customers in a service-provider network from other customers who seem to be on the same VLAN. You configure an asymmetric link from a tunnel port on a service-provider edge switch to an IEEE 802.1Q trunk port on the customer switch. Packets entering the tunnel port on the edge switch, already IEEE 802.1Q-tagged with the customer VLANs, are encapsulated with another layer of an IEEE 802.1Q tag (called the metro tag), containing a VLAN ID unique in the service-provider network, for each customer. The double-tagged packets go through the service-provider network, keeping the original customer VLANs separate from those of other customers. At the outbound interface, also a tunnel port, the metro tag is removed, and the original VLAN numbers from the customer network are retrieved.

Tunnel ports cannot be trunk ports or access ports and must belong to a VLAN unique to each customer.

For more information about tunnel ports, see [Chapter 14, “Configuring IEEE 802.1Q and Layer 2 Protocol Tunneling.”](#)

Switch Virtual Interfaces

A switch virtual interface (SVI) represents a VLAN of switch ports as one interface to the routing or bridging function in the system. Only one SVI can be associated with a VLAN, but you need to configure an SVI for a VLAN only when you wish to route between VLANs, fallback-bridge nonroutable protocols between VLANs, or to provide IP host connectivity to the switch. By default, an SVI is created for the default VLAN (VLAN 1) to permit remote switch administration. You cannot delete interface VLAN 1. Additional SVIs must be explicitly configured. In Layer 2 mode, SVIs provide IP host connectivity only to the system; in Layer 3 mode, you can configure routing across SVIs.

SVIs are created the first time that you enter the **vlan** interface configuration command for a VLAN interface. The VLAN corresponds to the VLAN tag associated with data frames on an ISL or IEEE 802.1Q encapsulated trunk or the VLAN ID configured for an access port. Configure a VLAN interface for each VLAN for which you want to route traffic, and assign it an IP address. For more information, see the [“Configuring IP Addressing on Layer 3 Interfaces”](#) section on page 31-4.

**Note**

When you create an SVI, it does not become active until it is associated with a physical port.

SVIs support routing protocols and bridging configurations. For more information about configuring IP routing, see [Chapter 31, “Configuring IP Unicast Routing,”](#) [Chapter 35, “Configuring IP Multicast Routing,”](#) and [Chapter 5, “Configuring Fallback Bridging.”](#)

**Note**

The IP base image, formerly known as the standard multilayer software image (SMI), supports static routing and the Routing Information Protocol (RIP). To use SVIs for full Layer 3 routing or for fallback bridging, you must have the IP services image, formerly known as the enhanced multilayer software image (EMI), installed on your switch.

Routed Ports

A routed port is a physical port that acts like a port on a router; it does not have to be connected to a router. A routed port is not associated with a particular VLAN, as is an access port. A routed port behaves like a regular router interface, except that it does not support VLAN subinterfaces. Routed ports can be configured with a Layer 3 routing protocol.

**Note**

The IP base image supports static routing and RIP; for more advanced routing, you must have the IP services image installed on your switch.

Configure routed ports by putting the interface into Layer 3 mode with the **no switchport** interface configuration command. Then assign an IP address to the port, enable routing, and assign routing protocol characteristics by using the **ip routing** and **router protocol** global configuration commands.

**Caution**

Entering a **no switchport** interface configuration command shuts the interface down and then re-enables it, which might generate messages on the device to which the interface is connected.

The number of routed ports and SVIs that you can configure is not limited by software; however, the interrelationship between this number and the number of other features being configured might have an impact on CPU utilization because of hardware limitations. For more information about feature combinations, see the [“Administering the Switch”](#) section on page 6-1.

For more information about IP unicast and multicast routing and routing protocols, see [Chapter 31, “Configuring IP Unicast Routing”](#) and [Chapter 35, “Configuring IP Multicast Routing.”](#)

EtherChannel Port Groups

EtherChannel port groups provide the ability to treat multiple switch ports as one switch port. These port groups act as a single logical port for high-bandwidth connections between switches or between switches and servers. An EtherChannel balances the traffic load across the links in the channel. If a link within the EtherChannel fails, traffic previously carried over the failed link changes to the remaining links. You can group multiple trunk ports into one logical trunk port, group multiple access ports into one logical access port, group multiple tunnel ports into one logical tunnel port, or group multiple routed ports into one logical routed port. Most protocols operate over either single ports or aggregated switch ports and do not recognize the physical ports within the port group. Exceptions are the DTP, the Cisco Discovery Protocol (CDP), and the Port Aggregation Protocol (PAgP), which operate only on physical ports.

When you configure an EtherChannel, you create a port-channel logical interface and assign an interface to the EtherChannel. For Layer 3 interfaces, you manually create the logical interface by using the **interface port-channel** global configuration command. For Layer 2 interfaces, the logical interface is dynamically created. For both Layer 3 and 2 interfaces, you manually assign an interface to the EtherChannel by using the **channel-group** interface configuration command. This command binds the physical and logical ports together. For more information, see [Chapter 30, “Configuring EtherChannels.”](#)

Power Over Ethernet Ports

Catalyst 3550 PoE-capable switch ports automatically supply power to these connected devices (if the switch senses that there is no power on the circuit):

- Cisco pre-standard powered devices (such as Cisco IP Phones and Cisco Aironet access points)
- IEEE 802.3af-compliant powered devices

A powered device can receive redundant power when it is connected to a PoE switch port and to an AC power source.

**Note**

PoE ports were previously referred to as inline power ports in earlier versions of the software configuration guide.

Supported Protocols and Standards

The switch uses these protocols and standards to support PoE:

- CDP with power consumption—The powered device notifies the switch of the amount of power it is consuming. The switch does not reply to the power-consumption messages. The switch can only supply power to or remove power from the PoE port.
- Cisco intelligent power management—The powered device and the switch negotiate through power-negotiation CDP messages for an agreed power-consumption level. The negotiation allows a high-power Cisco powered device, which consumes more than 7 W, to operate at its highest power mode. The powered device first boots up in low-power mode, consumes less than 7 W, and negotiates to obtain enough power to operate in high-power mode. The device changes to high-power mode only when it receives confirmation from the switch.

High-power devices can operate in low-power mode on switches that do not support power-negotiation CDP.

Before Release 12.1(22)EA2, Catalyst 3550 PoE-capable switches (without intelligent power management support) caused high-power powered devices that supported intelligent power management to operate in low-power mode. Devices in low-power mode are not fully functional.

Cisco intelligent power management is backward-compatible with CDP with power consumption; the switch responds according to the CDP message that it receives. CDP is not supported on third-party powered devices; therefore, the switch uses the IEEE classification to determine the power usage of the device.

- IEEE 802.3af—The major features of this standard are powered-device discovery, power administration, disconnect detection, and optional powered-device power classification. For more information, see the standard.

Powered-Device Detection and Initial Power Allocation

The switch detects a Cisco pre-standard or an IEEE-compliant powered device when the PoE-capable port is in the no-shutdown state, PoE is enabled (the default), and the connected device is not being powered by an AC adaptor.

After device detection, the switch determines the device power requirements based on its type:

- A Cisco pre-standard powered device does not provide its power requirement when the switch detects it, so the switch allocates 15.4 W as the initial allocation for power budgeting.

The initial power allocation is the maximum amount of power that a powered device requires. The switch initially allocates this amount of power when it detects and powers the powered device. As the switch receives CDP messages from the powered device and as the powered device negotiates power levels with the switch through CDP power-negotiation messages, the initial power allocation might be adjusted.

- For an IEEE device, the switch always allocates 15.4 W to the port. The switch does not display the IEEE class type in the **show power inline** privileged EXEC command output. Instead, it displays *n/a*.

The switch monitors and tracks requests for power and grants power only when it is available. The switch tracks its power budget (the amount of power available on the switch for PoE). The switch performs power-accounting calculations when a port is granted or denied power to keep the power budget up to date.

After power is applied to the port, the switch uses CDP to determine the *actual* power consumption requirement of the connected Cisco powered devices, and the switch adjusts the power budget accordingly. This does not apply to third-party PoE devices. The switch processes a request and either grants or denies power. If the request is granted, the switch updates the power budget. If the request is denied, the switch ensures that power to the port is turned off, generates a syslog message, and updates the LEDs. Powered devices can also negotiate with the switch for more power.

If the switch detects a fault caused by an undervoltage, overvoltage, overtemperature, oscillator-fault, or short-circuit condition, it turns off power to the port, generates a syslog message, and updates the power budget and LEDs.

Power Management Modes

The switch supports these PoE modes:

- **auto**—The switch automatically detects if the connected device requires power. If the switch discovers a powered device connected to the port and if the switch has enough power, it grants power, updates the power budget, turns on power to the port on a first-come, first-served basis, and updates the LEDs. For LED information, see the hardware installation guide.

If the switch has enough power for all the powered devices, they all come up. If enough power is available for all powered devices connected to the switch, power is turned on to all devices. If there is not enough available PoE, or if a device is disconnected and reconnected while other devices are waiting for power, it cannot be determined which devices are granted or are denied power.

If granting power would exceed the system power budget, the switch denies power, ensures that power to the port is turned off, generates a syslog message, and updates the LEDs. After power has been denied, the switch periodically rechecks the power budget and continues to attempt to grant the request for power.

If a device being powered by the switch is then connected to wall power, the switch might continue to power the device. The switch might continue to report that it is still powering the device whether the device is being powered by the switch or receiving power from an AC power source.

If a powered device is removed, the switch automatically detects the disconnect and removes power from the port. You can connect a nonpowered device without damaging it.

Use the **auto** setting on any PoE port. The auto mode is the default setting.

- **never**—The switch disables powered-device detection and never powers the PoE port even if an unpowered device is connected. Use this mode only when you want to make sure power is never applied to a PoE-capable port, making the port a data-only port.

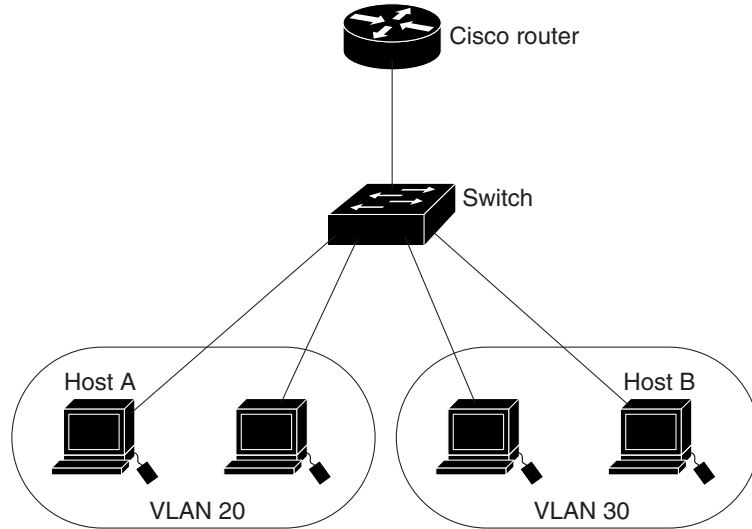
For information on configuring a PoE port, see the [“Configuring Power over Ethernet on the Catalyst 3550-24PWR Ports”](#) section on page 9-16.

Connecting Interfaces

Devices within a single VLAN can communicate directly through any switch. Ports in different VLANs cannot exchange data without going through a routing device or routed interface.

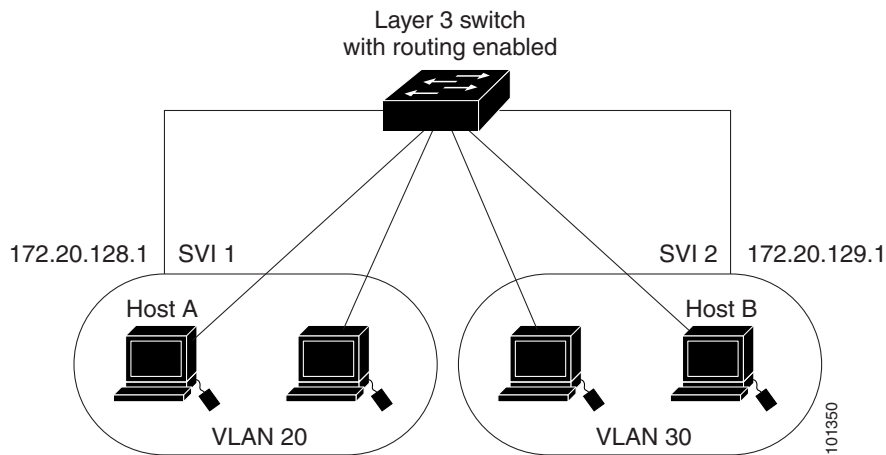
With a standard Layer 2 switch, ports in different VLANs have to exchange information through a router. In the configuration shown in [Figure 9-1](#), when Host A in VLAN 20 sends data to Host B in VLAN 30, it must go from Host A to the switch, to the router, back to the switch, and then to Host B.

Figure 9-1 Connecting VLANs with Layer 2 Switches



By using the Catalyst 3550 with routing enabled (as a Layer 3 switch), when you configure VLAN 20 and VLAN 30 each with an SVI to which an IP address is assigned, packets can be sent from Host A to Host B directly through the Catalyst 3550 switch with no need for an external router (Figure 9-2).

Figure 9-2 Connecting VLANs with the a Layer 3 Switch



The switch with the enhanced multilayer software image supports two methods of forwarding traffic between interfaces: routing and fallback bridging; the standard software image supports only basic routing (static routing and RIP). Whenever possible, to maintain high performance, forwarding is done by switch hardware. However, only IP version 4 packets with Ethernet II encapsulation can be routed in hardware. All other types of traffic can be fallback bridged by hardware.

- The routing function can be enabled on all SVIs and routed ports. The Catalyst 3550 switches route only IP traffic. When IP routing protocol parameters and address configuration are added to an SVI or routed port, any IP traffic received from these ports is routed. For more information, see [Chapter 31, “Configuring IP Unicast Routing,”](#) [Chapter 35, “Configuring IP Multicast Routing,”](#) and [Chapter 36, “Configuring MSDP.”](#)

- Fallback bridging forwards traffic that the switch with the enhanced multilayer software image does not route or traffic belonging to a nonroutable protocol, such as DECnet. Fallback bridging connects multiple VLANs into one bridge domain by bridging between two or more SVIs or routed ports. When configuring fallback bridging, you assign SVIs or routed ports to bridge groups with each SVI or routed port assigned to only one bridge group. All interfaces in the same group belong to the same bridge domain. For more information, see [Chapter 5, “Configuring Fallback Bridging.”](#)

Using the Interface Command

The switch supports these interface types:

- Physical ports—including switch ports and routed ports
- VLANs—switch virtual interfaces
- Port-channels—EtherChannel of interfaces

You can also configure a range of interfaces (see the [“Configuring a Range of Interfaces”](#) section on page 9-10).

To configure a physical interface (port), enter interface configuration mode, and specify the interface type, slot, and number.

- Type—Fast Ethernet (fastethernet or fa) for 10/100 Ethernet or Gigabit Ethernet (gigabitethernet or gi)
- Slot—The slot number on the switch (always 0 on this switch).
- Port number—The interface number on the switch. The port numbers always begin at 1, starting with the leftmost port when facing the front of the switch, for example, fastethernet0/1, fastethernet0/2. If there is more than one interface type (for example, 10/100 ports and Gigabit Ethernet ports), the port number restarts with the second interface type: gigabitethernet0/1, gigabitethernet0/2.

You can identify physical interfaces by physically checking the interface location on the switch. You can also use the Cisco IOS **show** privileged EXEC commands to display information about a specific interface or all the interfaces on the switch. The remainder of this chapter primarily provides physical interface configuration procedures.

Procedures for Configuring Interfaces

These general instructions apply to all interface configuration processes.

Step 1 Enter the **configure terminal** command at the privileged EXEC prompt:

```
Switch# configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Switch(config)#
```

Step 2 Enter the **interface** global configuration command. Identify the interface type and the number of the connector. In this example, Gigabit Ethernet interface 0/1 is selected:

```
Switch(config)# interface gigabitethernet0/1  
Switch(config-if)#
```



Note You do not need to add a space between the interface type and interface number. For example, in the preceding line, you can specify either **gigabitethernet 0/1**, **gigabitethernet0/1**, **gi 0/1**, or **gi0/1**.

Step 3 Follow each **interface** command with the interface configuration commands your particular interface requires. The commands you enter define the protocols and applications that will run on the interface. The commands are collected and applied to the interface when you enter another interface command or enter **end** to return to privileged EXEC mode.

You can also configure a range of interfaces by using the **interface range** or **interface range macro** global configuration commands. Interfaces configured in a range must be the same type and must be configured with the same feature options.

Step 4 After you configure an interface, verify its status by using the **show** privileged EXEC commands listed in the “[Monitoring and Maintaining the Interfaces](#)” section on page 9-21.

Enter the **show interfaces** privileged EXEC command to see a list of all interfaces on or configured for the switch. A report is provided for each interface that the device supports or for the specified interface:

Configuring a Range of Interfaces

You can use the **interface range** global configuration command to configure multiple interfaces with the same configuration parameters. When you enter the interface range configuration mode, all command parameters that you enter are attributed to all interfaces within that range until you exit this mode.

Beginning in privileged EXEC mode, follow these steps to configure a range of interfaces with the same parameters:

| | Command | Purpose |
|--------|---|---|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | interface range { <i>port-range</i> macro <i>macro_name</i> } | Specify the range of interfaces (VLANs or physical ports) to be configured, and enter interface range configuration mode. <ul style="list-style-type: none"> You can use the interface range command to configure up to five port ranges or a previously defined macro. The macro variable is explained in the “Configuring and Using Interface Range Macros” section on page 9-12. Each comma-separated <i>port-range</i> must consist of the same port type. You do not need to enter spaces before or after the comma. When you define a range, the space between the first port and the hyphen is required. |
| Step 3 | | You can now use the normal configuration commands to apply the configuration parameters to all interfaces in the range. |
| Step 4 | end | Return to privileged EXEC mode. |
| Step 5 | show interfaces [<i>interface-id</i>] | Verify the configuration of the interfaces in the range. |
| Step 6 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

When using the **interface range** global configuration command, note these guidelines:

- Valid entries for *port-range*:
 - **vlan** *vlan-ID* - *vlan-ID*, where VLAN ID is from 1 to 4094
 - **fastethernet** *slot*/*{first port}* - *{last port}*, where slot is 0
 - **gigabitethernet** *slot*/*{first port}* - *{last port}*, where slot is 0
 - **port-channel** *port-channel-number* - *port-channel-number*, where *port-channel-number* is from 1 to 64
- You must add a space between the interface numbers and the hyphen when using the **interface range** command. For example, the command **interface range fastethernet0/1 - 5** is a valid range; the command **interface range fastethernet0/1-5** is not a valid range.
- The **interface range** command works only with VLAN interfaces that have been configured with the **interface vlan** command (the **show running-config** privileged EXEC command displays the configured VLAN interfaces). VLAN interfaces not displayed by the **show running-config** command cannot be used with the **interface range** command.
- All interfaces in a range must be the same type; that is, all Fast Ethernet ports, all Gigabit Ethernet ports, all EtherChannel ports, or all SVIs.

This example shows how to use the **interface range** global configuration command to a range of interfaces:

```
Switch# configure terminal
Switch(config)# interface range fastethernet0/1 - 5
Switch(config-if-range)# no shutdown
Switch(config-if-range)#
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to up
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet0/3, changed state to up
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet0/4, changed state to up
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet0/5, changed state to up
*Oct 6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/05,
changed state to up
*Oct 6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed
state to up
*Oct 6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/4, changed
state to up
```

This example shows how to use a comma to add different interface type strings to the range to enable all Fast Ethernet and Gigabit Ethernet interfaces:

```
Switch# configure terminal
Switch(config)# interface range fastethernet0/1 - 3, gigabitethernet0/1 - 2
Switch(config-if-range)# no shutdown
Switch(config-if-range)#
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet0/3, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
*Oct 6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/ 1,
changed state to up
*Oct 6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/ 2,
changed state to up
*Oct 6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/ 3,
changed state to up
```

If you enter multiple configuration commands while you are in interface range mode, each command is executed as it is entered. The commands are not batched together and executed after you exit interface range mode. If you exit interface range configuration mode while the commands are being executed, some commands might not be executed on all interfaces in the range. Wait until the command prompt reappears before exiting interface range configuration mode.

Configuring and Using Interface Range Macros

You can create an interface range macro to automatically select a range of interfaces for configuration. Before you can use the **macro** keyword in the **interface range macro** global configuration command string, you must use the **define interface-range** global configuration command to define the macro.

Beginning in privileged EXEC mode, follow these steps to define an interface range macro:

| | Command | Purpose |
|--------|---|---|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | define interface-range <i>macro_name</i> <i>interface-range</i> | Define the interface-range macro, and save it in NVRAM. <ul style="list-style-type: none"> • The <i>macro_name</i> is a 32-character maximum character string. • A macro can contain up to five comma-separated interface ranges. You do not need to enter spaces before or after the comma. • Each <i>interface-range</i> must consist of the same port type. |
| Step 3 | interface range macro <i>macro_name</i> | Select the interface range to be configured using the values saved in the interface-range macro called <i>macro_name</i> . You can now use the normal configuration commands to apply the configuration to all interfaces in the defined macro. |
| Step 4 | end | Return to privileged EXEC mode. |
| Step 5 | show running-config include define | Show the defined interface range macro configuration. |
| Step 6 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Use the **no define interface-range** *macro_name* global configuration command to delete a macro.

When using the **define interface-range** global configuration command, note these guidelines:

- Valid entries for *interface-range*:
 - **vlan** *vlan-ID* - *vlan-ID*, where VLAN ID is from 1 to 4094
 - **fastethernet** *slot*{*first port*} - {*last port*}, where slot is **0**
 - **gigabitethernet** *slot*{*first port*} - {*last port*}, where slot is **0**
 - **port-channel** *port-channel-number* - *port-channel-number*, where *port-channel-number* is from 1 to 64.
- You must add a space between the interface numbers and the hyphen when entering an *interface-range*. For example, **fastethernet0/1 - 5** is a valid range; **fastethernet0/1-5** is not a valid range.
- The VLAN interfaces (SVIs) must have been configured with the **interface vlan** command. The **show running-config** privileged EXEC command displays the configured VLAN interfaces. VLAN interfaces not displayed by the **show running-config** command cannot be used as *interface-ranges*.

- All interfaces in a range must be the same type; that is, all Fast Ethernet ports, all Gigabit Ethernet ports, all EtherChannel ports, or all VLANs, but you can combine multiple interface types in a macro.

This example shows how to define an interface-range macro named *enet_list* to select Fast Ethernet ports 1 to 4 and to verify the macro configuration:

```
Switch# configure terminal
Switch(config)# define interface-range enet_list fastethernet0/1 - 4
Switch(config)# end
Switch# show running-config | include define
define interface-range enet_list FastEthernet0/1 - 4
```

This example shows how to create a multiple-interface macro named *macro1*:

```
Switch# configure terminal
Switch(config)# define interface-range macro1 gigabitethernet0/1 - 2, fastethernet0/5 - 7
Switch(config)# end
Switch#
```

This example shows how to enter interface range configuration mode for the interface-range macro *enet_list*:

```
Switch# configure terminal
Switch(config)# interface range macro enet_list
Switch(config-if-range)#
```

This example shows how to delete the interface-range macro *enet_list* and to verify that it has been deleted.

```
Switch# configure terminal
Switch(config)# no define interface-range enet_list
Switch# show run | include define
```

Configuring Ethernet Interfaces

These sections describe the default interface configuration and the optional features that you can configure on most physical interfaces:

- [Default Ethernet Interface Configuration, page 9-14](#)
- [Configuring Interface Speed and Duplex Mode, page 9-14](#)
- [Configuring Power over Ethernet on the Catalyst 3550-24PWR Ports, page 9-16](#)
- [Configuring IEEE 802.3x Flow Control, page 9-17](#)
- [Adding a Description for an Interface, page 9-18](#)



Caution

If the interface is in Layer 3 mode, after entering interface configuration mode, you must enter the **switchport** interface configuration command without any parameters to put the interface into Layer 2 mode. This shuts down the interface and then re-enables it, which might generate messages on the device to which the interface is connected. Furthermore, when you use this command to put the interface into Layer 2 mode, you are deleting any Layer 3 characteristics configured on the interface.

Default Ethernet Interface Configuration

Table 9-1 shows the Ethernet interface default configuration. For more details on the VLAN parameters listed in the table, see Chapter 11, “Configuring VLANs.” For details on controlling traffic to the port, see Chapter 21, “Configuring Port-Based Traffic Control.”

Table 9-1 Default Ethernet Interface Configuration

| Feature | Default Setting |
|--|---|
| Operating mode | Layer 2 or switching mode (switchport command). |
| Allowed VLAN range | VLANs 1 – 4094. |
| Default VLAN (for access ports) | VLAN 1. |
| Native VLAN (for IEEE 802.1Q trunks) | VLAN 1. |
| VLAN trunking | Switchport mode dynamic desirable (supports DTP). |
| Port enable state | All ports are enabled. |
| Port description | None defined. |
| Speed | Autonegotiate. |
| Duplex mode | Autonegotiate. |
| Flow control | Flow control is set to <i>off</i> for receive and <i>desired</i> for send for Gigabit Ethernet ports. For 10/100 Mb/s ports, send is always <i>off</i> . |
| Power over Ethernet (supported only on the Catalyst 3550-24PWR switch) | Enabled (auto). |
| EtherChannel (PAgP) | Disabled on all Ethernet ports. See Chapter 30, “Configuring EtherChannels.” |
| Port blocking (unknown multicast and unknown unicast traffic) | Disabled (not blocked). See the “Configuring Port Blocking” section on page 21-6. |
| Broadcast, multicast, and unicast storm control | Disabled. See the “Default Storm Control Configuration” section on page 21-3. |
| Protected port | Disabled. See the “Configuring Protected Ports” section on page 21-5. |
| Port security | Disabled. See the “Default Port Security Configuration” section on page 21-9. |
| Port Fast | Disabled. |

Configuring Interface Speed and Duplex Mode

Ethernet interfaces on the switch operate in 10, 100, or 1000 Mbps and in either full- or half-duplex mode. In full-duplex mode, two stations can send and receive at the same time. When packets can flow in both directions simultaneously, effective Ethernet bandwidth doubles to 20 Mbps for 10-Mbps interfaces, to 200 Mbps for Fast Ethernet interfaces, and to 2 Gbps for Gigabit interfaces. Full-duplex communication is often an effective solution to collisions, which are major constrictions in Ethernet networks. Normally, 10-Mbps ports operate in half-duplex mode, which means that stations can either receive or send.

You can configure interface speed on Fast Ethernet (10/100-Mbps) and Gigabit Ethernet (10/100/1000-Mbps) interfaces; you cannot configure speed on Gigabit Interface Converter (GBIC) interfaces. You can configure duplex mode on any Fast Ethernet or Gigabit Ethernet interfaces that are not set to autonegotiate; you cannot configure duplex mode on GBIC interfaces.

**Note**

You cannot configure speed or duplex mode on Gigabit Interface Converter (GBIC) ports, but for certain types of GBICs, you can configure speed to not negotiate (**nonegotiate**) if connected to a device that does not support autonegotiation.

These sections describe how to configure the interface speed and duplex mode:

- [Configuration Guidelines, page 9-15](#)
- [Setting the Interface Speed and Duplex Parameters, page 9-15](#)

Configuration Guidelines

When configuring an interface speed and duplex mode, note these guidelines:

- If both ends of the line support autonegotiation, we highly recommend the default setting of **autonegotiation**.
- If one interface supports autonegotiation and the other end does not, configure duplex and speed on both interfaces; do not use the **auto** setting on the supported side.
- 100BASE-FX ports operate only at 100 Mbps in either full- or half-duplex mode and do not support autonegotiation.
- GigaStack-to-GigaStack cascade connections operate in half-duplex mode, and GigaStack-to-GigaStack point-to-point connections operate in full-duplex mode.
- When STP is enabled and a port is reconfigured, the switch can take up to 30 seconds to check for loops. The port LED is amber while STP reconfigures.

**Caution**

Changing the interface speed and duplex mode configuration might shut down and re-enable the interface during the reconfiguration.

Setting the Interface Speed and Duplex Parameters

Beginning in privileged EXEC mode, follow these steps to set the speed and duplex mode for a physical interface:

| | Command | Purpose |
|--------|--------------------------------------|--|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | interface <i>interface-id</i> | Specify the physical interface to be configured, and enter interface configuration mode. |

| | Command | Purpose |
|--------|--|--|
| Step 3 | <code>speed {10 100 1000 auto [10 100 1000] nonegotiate}</code> | Enter the appropriate speed parameter for the interface, or enter auto or nonegotiate . Note The 1000 keyword is available only for 10/100/1000 Mbps ports. 100BASE-FX ports operate only at 100 Mbps. GBIC module ports operate only at 1000 Mbps. The nonegotiate keyword is available only for 1000BASE-SX, -LX, and -ZX GBIC ports. If you use the 10 , 100 , or 1000 keywords with the auto keyword, the port only autonegotiates at the specified speeds. |
| Step 4 | <code>duplex {auto full half}</code> | Enter the duplex parameter for the interface. Note 100BASE-FX ports operate only in full-duplex mode. This keyword is not available on GBIC ports. |
| Step 5 | <code>end</code> | Return to privileged EXEC mode. |
| Step 6 | <code>show interfaces interface-id</code> | Display the interface speed and duplex mode configuration. |
| Step 7 | <code>copy running-config startup-config</code> | (Optional) Save your entries in the configuration file. |

Use the **no speed** and **no duplex** interface configuration commands to return the interface to the default speed and duplex settings (autonegotiate). To return all interface settings to the defaults, use the **default interface interface-id** interface configuration command.

This example shows how to set the interface speed to 10 Mbps and the duplex mode to half:

```
Switch# configure terminal
Switch(config)# interface fastethernet0/3
Switch(config-if)# speed 10
Switch(config-if)# duplex half
```

Configuring Power over Ethernet on the Catalyst 3550-24PWR Ports

The Catalyst 3550-24PWR switch automatically supplies Power over Ethernet (PoE) to connected Cisco IP Phones, Cisco Aironet Access Points, and IEEE-compliant powered devices if it senses *no* power on the circuit. If there is power on the circuit, the switch does not supply it.



Note

PoE ports were previously referred to as inline power ports in earlier versions of the software configuration guide.

For information about configuring a switch port to forward IP voice traffic to and from connected Cisco IP Phones, see the [“Configuring a Port to Connect to a Cisco 7960 IP Phone”](#) section on page 13-3.

For information about configuring the switch for certain IEEE-compliant powered devices that require multiple reloads during initialization, see the **power inline** interface configuration command in the command reference for this release.

Beginning in privileged EXEC mode, follow these steps to enable PoE on a PoE-capable port:

| | Command | Purpose |
|--------|---|--|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | interface <i>interface</i> | Specify the port to be configured, and enter interface configuration mode. |
| Step 3 | power inline auto | Enable powered-device detection. If enough power is available, automatically allocate power to the PoE port after device detection. This is the default. |
| Step 4 | end | Return to privileged EXEC mode. |
| Step 5 | show power inline <i>interface</i> | Verify the change. |
| Step 6 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

To disable PoE on a port, use the **power inline never** interface configuration command.

**Note**

If a port has a Cisco powered device connected to it, you should not use the **power inline never** command to configure the port. A false link-up can occur on the port, placing the port into an error-disabled state.

Configuring IEEE 802.3x Flow Control

Flow control enables connected Ethernet ports to control traffic rates during congestion by allowing congested nodes to pause link operation at the other end. If one port experiences congestion and cannot receive any more traffic, it notifies the other port to stop sending until the condition clears. When the local device detects any congestion at its end, it can notify the link partner or the remote device of the congestion by sending a pause frame. Upon receipt of a pause frame, the remote device stops sending any data packets, which prevents any loss of data packets during the congestion period.

**Note**

You must not configure both IEEE 802.3z flowcontrol and quality of service (QoS) on a switch. Before configuring flowcontrol on an interface, use the **no mls qos** global configuration command to disable QoS on the switch.

Flow control can be implemented in two forms, symmetric and asymmetric. The symmetric implementation is suitable for point-to-point links, and asymmetric is suitable for hub-to-end node connections, where it is desirable for the hub to pause the end system, but not vice-versa. You use the **flowcontrol** interface configuration command to set the interface's ability to **receive** and **send** pause frames to **on**, **off**, or **desired**. The default state for Gigabit Ethernet ports is **receive off** and **send desired**. The default state for Fast Ethernet ports is **receive off** and **send off**.

**Note**

On Catalyst 3550 switches, Gigabit Ethernet ports are capable of receiving and sending pause frames; Fast Ethernet ports can only receive pause frames. Therefore, for Fast Ethernet ports, only the conditions described with **send off** are applicable.

These rules apply to flow control settings on the device:

- **receive on** (or **desired**) and **send on**: Flow control operates in both directions; both the local and the remote devices can send pause frames to show link congestion.

- **receive on** (or **desired**) and **send desired**: The port can receive pause frames and can send pause frames if the attached device supports flow control.
- **receive on** (or **desired**) and **send off**: The port cannot send pause frames but can operate with an attached device that is required to or can send pause frames; the port can receive pause frames.
- **receive off** and **send on**: The port sends pause frames if the remote device supports flow control but cannot receive pause frames from the remote device.
- **receive off** and **send desired**: The port cannot receive pause frames but can send pause frames if the attached device supports flow control.
- **receive off** and **send off**: Flow control does not operate in either direction. In case of congestion, no indication is given to the link partner, and no pause frames are sent or received by either device.

**Note**

For details on the command settings and the resulting flow control resolution on local and remote ports, see the **flowcontrol** interface configuration command in the command reference for this release.

Beginning in privileged EXEC mode, follow these steps to configure flow control on an interface:

| | Command | Purpose |
|--------|---|--|
| Step 1 | configure terminal | Enter global configuration mode |
| Step 2 | no mls qos | Disable QoS on the switch. |
| Step 3 | interface <i>interface-id</i> | Specify the physical port to be configured, and enter interface configuration mode. |
| Step 4 | flowcontrol { receive send } { on off desired } | Configure the flow control mode for the port. Note The send keyword is not available for 10/100 Mbps ports. |
| Step 5 | end | Return to privileged EXEC mode. |
| Step 6 | show interfaces <i>interface-id</i> | Verify the interface flow control settings. |
| Step 7 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

To disable flow control, use the **flowcontrol receive off** and **flowcontrol send off** interface configuration commands.

This example shows how to turn off all flow control on an interface:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# flowcontrol receive off
Switch(config-if)# flowcontrol send off
Switch(config-if)# end
```

Adding a Description for an Interface

You can add a description about an interface to help you remember its function. The description appears in the output of these commands: **show configuration**, **show running-config**, and **show interfaces**.

Beginning in privileged EXEC mode, follow these steps to add a description for an interface:

| | Command | Purpose |
|--------|---|---|
| Step 1 | configure terminal | Enter global configuration mode |
| Step 2 | interface <i>interface-id</i> | Specify the interface for which you are adding a description, and enter interface configuration mode. |
| Step 3 | description <i>string</i> | Add a description (up to 240 characters) for an interface. |
| Step 4 | end | Return to privileged EXEC mode. |
| Step 5 | show interfaces <i>interface-id</i> description or show running-config | Verify your entry. |
| Step 6 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Use the **no description** interface configuration command to delete the description.

This example shows how to add a description on an interface and to verify the description:

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet0/4
Switch(config-if)# description Connects to Marketing
Switch(config-if)# end
Switch# show interfaces fastethernet0/4 description
Interface Status      Protocol Description
Fa0/4      up          down      Connects to Marketing
```

Configuring Layer 3 Interfaces

The switch supports three types of Layer 3 interfaces:

- SVIs: You should configure SVIs for any VLANs for which you want to route traffic. SVIs are created when you enter a VLAN ID following the **interface vlan** global configuration command. To delete an SVI, use the **no interface vlan** global configuration command. You cannot delete interface VLAN 1.



Note When you create an SVI, it does not become active until it is associated with a physical port. For information about assigning Layer 2 ports to VLANs, see [Chapter 11, “Configuring VLANs.”](#)

- Layer 3 EtherChannel ports: EtherChannel interfaces made up of routed ports. EtherChannel port interfaces are described in [Chapter 30, “Configuring EtherChannels.”](#)
- Routed ports: Routed ports are physical ports configured to be in Layer 3 mode by using the **no switchport** interface configuration command.

**Note**

A Layer 3 switch can have an IP address assigned to each routed port and SVI. The number of routed ports and SVIs that you can configure is not limited by software; however, the interrelationship between this number and the number of other features being configured might have an impact on CPU utilization because of hardware limitations. For more information about feature combinations, see the [“Optimizing System Resources for User-Selected Features”](#) section on page 6-26.

All Layer 3 interfaces require an IP address to route traffic. The following procedure shows how to configure an interface as a Layer 3 interface and how to assign an IP address to an interface.

**Note**

If the physical port is in Layer 2 mode (the default), you must enter the **no switchport** interface configuration command to put the interface into Layer 3 mode. Entering a **no switchport** command disables and then re-enables the interface, which might generate messages on the device to which the interface is connected.

Beginning in privileged EXEC mode, follow these steps to configure a Layer 3 interface:

| | Command | Purpose |
|--------|--|--|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | interface {{fastethernet gigabitethernet} <i>interface-id</i> {vlan <i>vlan-id</i> } {port-channel <i>port-channel-number</i> } | Specify the interface to be configured as a Layer 3 interface, and enter interface configuration mode. |
| Step 3 | no switchport | For physical ports only, enter Layer 3 mode. |
| Step 4 | ip address <i>ip_address subnet_mask</i> | Configure the IP address and IP subnet. |
| Step 5 | no shutdown | Enable the interface. |
| Step 6 | end | Return to privileged EXEC mode. |
| Step 7 | show interfaces [<i>interface-id</i>] show ip interface [<i>interface-id</i>] show running-config interface [<i>interface-id</i>] | Verify the configuration. |
| Step 8 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

To remove an IP address from an interface, use the **no ip address** interface configuration command.

This example shows how to configure an interface as a routed port and to assign it an IP address:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet0/2
Switch(config-if)# no switchport
Switch(config-if)# ip address 192.20.135.21 255.255.255.0
Switch(config-if)# no shutdown
Switch(config-if)# end
```

Monitoring and Maintaining the Interfaces

You can perform the tasks in these sections to monitor and maintain interfaces:

- [Monitoring Interface and Controller Status, page 9-21](#)
- [Clearing and Resetting Interfaces and Counters, page 9-22](#)
- [Shutting Down and Restarting the Interface, page 9-22](#)

Monitoring Interface and Controller Status

Commands entered at the privileged EXEC prompt display information about the interface, including the version of the software and the hardware, the controller status, and statistics about the interfaces. [Table 9-2](#) lists some of these interface monitoring commands. (You can display the full list of **show** commands by using the **show ?** command at the privileged EXEC prompt.) These commands are fully described in the *Cisco IOS Interface Command Reference, Release 12.2*.

Table 9-2 *show Commands for Interfaces*

| Command | Purpose |
|--|---|
| show interfaces [<i>interface-id</i>] | Display the status and configuration of all interfaces or a specific interface. |
| show interfaces [<i>interface-id</i>] capabilities [<i>module</i> { <i>module-number</i> }] | Display the capabilities of an interface. The module number is always 0. If you enter an interface ID, the module keyword is not visible. |
| show interfaces <i>interface-id</i> status [err-disabled] | Display interface status or a list of interfaces in error-disabled state. |
| show interfaces [<i>interface-id</i>] switchport | Display administrative and operational status of switching (nonrouting) ports. You can use this command to determine if a port is in routing or switching mode. |
| show interfaces [<i>interface-id</i>] description | Display the description configured on an interface or all interfaces and the interface status. |
| show ip interface [<i>interface-id</i>] | Display the usability status of all interfaces configured for IP or the specified interface. |
| show interfaces transceiver properties | (Optional) Display speed, duplex, and inline power settings on the interface. |
| show running-config interface [<i>interface-id</i>] | Display the running configuration in RAM for the interface. |
| show version | Display the hardware configuration, software version, the names and sources of configuration files, and the boot images. |

For examples of output displays and definitions of output fields for the **show interfaces** privileged EXEC command, see the command reference for this release.

Clearing and Resetting Interfaces and Counters

Table 9-3 lists the privileged EXEC mode **clear** commands that you can use to clear counters and reset interfaces.

Table 9-3 Clear Commands for Interfaces

| Command | Purpose |
|---|--|
| clear counters [<i>interface-id</i>] | Clear interface counters. |
| clear interface <i>interface-id</i> | Reset the hardware logic on an interface. |
| clear line [<i>number</i> console 0 vtty number] | Reset the hardware logic on an asynchronous serial line. |

To clear the interface counters shown by the **show interfaces** privileged EXEC command, use the **clear counters** privileged EXEC command. The **clear counters** command clears all current interface counters from the interface unless optional arguments are specified to clear only a specific interface type from a specific interface number.



Note

The **clear counters** privileged EXEC command does not clear counters retrieved by using Simple Network Management Protocol (SNMP), but only those seen with the **show interface** privileged EXEC command.

This example shows how to clear and reset the counters on an interface:

```
Switch# clear counters fastethernet0/5
Clear "show interface" counters on this interface [confirm] y
Switch#
*Sep 30 08:42:55: %CLEAR-5-COUNTERS: Clear counter on interface FastEthernet0/5
by vty1 (171.69.115.10)
```

Use the **clear interface** or **clear line** privileged EXEC command to clear and reset an interface or serial line. Under most circumstances, you do not need to clear the hardware logic on interfaces or serial lines.

This example shows how to clear and reset an interface:

```
Switch# clear interface fastethernet0/5
```

Shutting Down and Restarting the Interface

Shutting down an interface disables all functions on the specified interface and marks the interface as unavailable on all monitoring command displays. This information is communicated to other network servers through all dynamic routing protocols. The interface is not mentioned in any routing updates.

Beginning in privileged EXEC mode, follow these steps to shut down an interface:

| | Command | Purpose |
|--------|---|--|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | interface { vlan <i>vlan-id</i> } { fastethernet gigabitethernet } <i>interface-id</i> } { port-channel <i>port-channel-number</i> } | Select the interface to be configured. |
| Step 3 | shutdown | Shut down an interface. |

| | Command | Purpose |
|--------|----------------------------|---------------------------------|
| Step 4 | end | Return to privileged EXEC mode. |
| Step 5 | show running-config | Verify your entry. |

Use the **no shutdown** interface configuration command to restart the interface.

This example shows how to shut down an interface:

```
Switch# configure terminal
Switch(config)# interface fastethernet0/5
Switch(config-if)# shutdown
Switch(config-if)#
*Sep 30 08:33:47: %LINK-5-CHANGED: Interface FastEthernet0/5, changed state to a
administratively down
```

This example shows how to re-enable an interface:

```
Switch# configure terminal
Switch(config)# interface fastethernet0/5
Switch(config-if)# no shutdown
Switch(config-if)#
*Sep 30 08:36:00: %LINK-3-UPDOWN: Interface FastEthernet0/5, changed state to up
```

To verify that an interface is disabled, enter the **show interfaces** privileged EXEC command. A disabled interface is shown as *administratively down* in the **show interface** command display.

